NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

Priority: IST-2004-2.4.7 – "Semantic-based knowledge and content systems"

D7.6.2 Second Prototype of the Fisheries Stock Depletion Assessment System (FSDAS)

Deliverable Co-ordinator: **Michael Erdmann**

Deliverable Co-ordinating Institution: **Ontoprise GmbH (ONTO)**

Other Authors:
Claudio Baldassarre (FAO)
Caterina Caracciolo (FAO)
Aldo Gangemi (CNR)
Germán Herrero Cárcel (ATOS)
Tomás Pariente Lobo (ATOS)
Wolfgang Schoch (ONTO)

This document describes the implementation of the second prototype of the Fisheries Stock Depletion Assessment System (FSDAS). It explains the architecture of FSDAS, the networked ontologies used, and provides an insight into the design and deployment of the software.
FSDAS is a web based application making use of ontological knowledge to help fishery experts to access different data sources about catch data and other related information.

Document Identifier:	NEON/2009/D7.6.2/v1.0	Date due:	February 28th, 2009
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	February 28th, 2009
Project start date:	March 1, 2006	Version:	v1.0
Project duration:	4 years	State:	Final
		Distribution:	Public

NeOn Consortium

This document is a part of the NeOn research project funded by the IST Programme of the Commission of the European Communities by the grant number IST-2005-027595. The following partners are involved in the project:

Open University (OU) – Coordinator Knowledge Media Institute – KMi Berrill Building, Walton Hall Milton Keynes, MK7 6AA United Kingdom Contact person: Martin Dzbor, Enrico Motta E-mail address: {m.dzbor, e.motta} @open.ac.uk	Universität Karlsruhe – TH (UKARL) Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB Englerstrasse 11 D-76128 Karlsruhe, Germany Contact person: Peter Haase E-mail address: pha@aifb.uni-karlsruhe.de
Universidad Politécnica de Madrid (UPM) Campus de Montegancedo 28660 Boadilla del Monte Spain Contact person: Asunción Gómez Pérez E-mail address: asun@fi.upm.es	Software AG (SAG) Uhlandstrasse 12 64297 Darmstadt Germany Contact person: Walter Waterfeld E-mail address: walter.waterfeld@softwareag.com
Intelligent Software Components S.A. (ISOCO) Calle de Pedro de Valdivia 10 28006 Madrid Spain Contact person: Jesús Contreras E-mail address: jcontreras@isoco.com	Institut 'Jožef Stefan' (JSI) Jamova 39 SI-1000 Ljubljana Slovenia Contact person: Marko Grobelnik E-mail address: marko.grobelnik@ijs.si
Institut National de Recherche en Informatique et en Automatique (INRIA) ZIRST – 655 avenue de l'Europe Montbonnot Saint Martin 38334 Saint-Ismier France Contact person: Jérôme Euzenat E-mail address: Jerome.euzenat@inrialpes.fr	University of Sheffield (USFD) Dept. of Computer Science Regent Court 211 Portobello street S14DP Sheffield United Kingdom Contact person: Hamish Cunningham E-mail address: hamish@dcs.shef.ac.uk
Universität Koblenz-Landau (UKO-LD) Universitätsstrasse 1 56070 Koblenz Germany Contact person: Steffen Staab E-mail address: staab@uni-koblenz.de	Consiglio Nazionale delle Ricerche (CNR) Institute of cognitive sciences and technologies Via S. Martino della Battaglia, 44 - 00185 Roma-Lazio, Italy Contact person: Aldo Gangemi E-mail address: aldo.gangemi@istc.cnr.it
Ontoprise GmbH. (ONTO) Amalienbadstr. 36 (Raumfabrik 29) 76227 Karlsruhe Germany Contact person: Jürgen Angele E-mail address: angele@ontoprise.de	Food and Agriculture Organization of the United Nations (FAO) Viale delle Terme di Caracalla 1 00100 Rome Italy Contact person: Marta Iglesias E-mail address: marta.iglesias@fao.org
Atos Origin S.A. (ATOS) Calle de Albarracín, 25 28037 Madrid, Spain Contact person: Tomás Pariente Lobo E-mail address: tomas.parientelobo@atosorigin.com	Laboratorios KIN, S.A. (KIN) C/Ciudad de Granada, 123 08018 Barcelona, Spain Contact person: Antonio López E-mail address: alopez@kin.es

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed writing parts of this document:

ATOS, CNR, FAO, ONTO

Change Log

Version	Date	Amended by	Changes
0.1	2009-02-13	Michael Erdmann	Initial version of deliverable document
0.2	2009-03-02	Michael, Claudio	Added text for ontologies and data sources, architecture and goals/intro
0.21		Claudio	Updated user descriptions
0.3	2009-03-16	Michael	Updated use case and architecture sections
0.31	2009-03-16	Caterina	Update of information sources
0.4	2009-03-21	Michael	Updated ontology and architecture sections. Added references
0.5	2009-04-07	Tomás, Claudio	Executive summary, introduction and conclusion
0.6	2009-04-08	German	Update sections 4.3.2 and 4.4
0.7	2009-04-09	Michael	Review and fine tuning of new sections
0.8	2009-04-09	Wolgang, Michael	Description of the data-tier
0.9	2009-04-10	Aldo	Added Ontology design chapter
0.95	2009-04-13	Germán, Tomás	Appendix 2 added
1.0	2009-04-14	Michael	Finishing touches
1.1	2009-04-15	Michael	Consideration of QA comments
1.2	2009-04-15	Germán, Tomás	Consideration of QA comments
1.3	2009-04-16	Michael	Rephrasing of some unclear paragraphs

Executive Summary

This document describes the implementation of the second prototype of the Fisheries Stock Depletion Assessment System (FSDAS).

The evolution of the FSDAS prototype from the previous version is significant. The first prototype was delivered as an Eclipse plugin, and did not use networked but only single and not connected ontologies. On the contrary, the current version of the prototype is Web-based, using OntoBroker in the back-end, using a real network of ontologies, and covering a range of functionalities more focused on the FAO real needs.

The document explains the architecture of the FSDAS, the networked ontologies used, and provides an insight into the design and deployment of the software developed.

It is foreseen that during the evaluation of the prototype some enhancements will be delivered. This document goes together with the software developed for this second FSDAS prototype as a part of task 7.6.

Table of Contents

List of Figures

List of Tables

1 Introduction

This document describes the implementation of the second prototype of the Fisheries Stock Depletion Assessment System (FSDAS). The document explains the architecture of FSDAS, the networked ontologies used, and provides an insight into the design and deployment of the software developed.

The second iteration of FSDAS makes use of a knowledge base consisting of networked ontologies that provides information about fish stocks and associated knowledge about other types of entities, such as geographical information or information about gear types, vessels, etc. The knowledge base is populated with original data available from the FAO fisheries department. The conceptual structure of the KB is based on the ontologies reengineered from the RTMS database of taxonomic knowledge.

1.1 Differences to the first FSDAS Prototype

The evolution of the FSDAS prototype from the previous version is significant. The major changes are the following:

- The first prototype was delivered as an Eclipse plugin,

  In contrast, the current version of the prototype is using a Web-based interface with a server-side that includes OntoBroker in the back-end. This architectural change implied a big effort, but it fits better with the FAO needs and simplifies the deployment of the software.

- The first prototype used only F-logic ontologies.

  The second prototype is using a combination of OWL-DL ontologies and extracted facts that are translated into RDF triples. The different languages are completely transparent to the user.

- The ontologies used in the first prototype were not connected (networked) at all.

  The second prototype is using several ontologies that form a real network and provide different kinds of knowledge to the overall knowledge base.

- The first prototype covered a limited set of the most important requirements.

  For the second prototype we are focussing on the real information needs for the users from the fisheries departments and thus expect better usefulness of FSDAS for the domain users.

It is foreseen that during the evaluation of the prototype some more enhancements will be delivered and reported back along with the evaluation report. This document goes together with the software developed for this prototype as a part of task 7.6.

1.2 Structure of the document

In this document we first present in Section 2 the main goals and requirements of the FSDAS, along with the data sources used. Section 3 is devoted to explaining the networked ontologies used by the FSDAS. The design and architecture of the current release of the FSDAS prototype is explained in Section 4. After some conclusions and the list of

references (Sections 5 and 6), we include as appendixes the installation guidelines and a brief user manual of the graphical user interface of the application.

2 Goals of FSDAS

FSDAS is an ontology-driven Fisheries Stock Depletion Assessment System. It embodies a knowledge base holding relevant information about fish stocks and associated knowledge about other types of entities. This knowledge consists of mainly two different types:

- Taxonomic knowledge stemming from multiple ontologies reengineered from the RTMS database of taxonomic terms.
- Factual information about many different entities stemming from XML fact sheets published by the FAO fishery department.

The current implementation of FSDAS is the second iteration and improves the feature-set of the first prototype of FSDAS [NeOn D7.6.1] in a number of dimensions, e.g.

- Flexible query answering approach
- Access to real data
- Combination of keyword search and formal query mechanism
- Web-based user interface
- Employment of multiple, networked ontologies

The original requirements document [NeOn D7.1.1] and the original design document [NeOn D7.5.1] for an FSDAS system define a number of use cases and requirements to inform the design and development activities. Some of them have been implemented in the first prototype, which has been evaluated in [NeOn D7.7.1]. This evaluation mainly points to weaknesses in the presentation of the functionality and the general look and feel. These insights were considered when selecting the most relevant use cases for the second iteration (cf. Section 2.2)

2.1 Target User Group

The users of the FSDAS application will mainly be officers in FAO's Fisheries and Aquaculture Department FI (including Fishery Policy and Planning Division, Fisheries Resources Division, Fishery industry Division, as well as the Fishery Information, Data and Statistics Unit) attached to FAO headquarters and regional and sub-regional offices around the world. In this sense, the user classes are all fisheries experts within some domain of fisheries: Fisheries Scientists, Fisheries Managers, Marine Biologists, Oceanographers, Fisheries economists, Fisheries legal experts, Fisheries engineers, Fisheries policy makers.

Currently, FAO fisheries experts producing fisheries and stock assessments are forced to access numerous different information systems in order to locate the data they need to create their reports. This seriously hampers, both, searching for data and citing it. In addition, the necessity of searching through numerous systems means that reports may possibly neglect some data source one year, and use it the next, creating possible inconsistencies across years.

2.2 Use cases

The use-cases included in the second version of FSDAS represent a subset of the ones originally required. This decision is a consequence of a development process more focused on few specialized cases, given the time frame dedicated to their development in the scope of having the application redeployed in a web environment.

We elicited what is the minimum set of use-cases to include, such that FSDAS was faithful to its characteristics given by the initial core objectives described in [NeOn D.7.11]. The agreed list of use cases is presented in Table 1.

Table 1: Use Cases representative of the functionalities included in FSDAS v2

Use Case	Comment	Priority		First Prototype
		High	Low	
Search ontological resource in ontology	Search for ontology content as data value e.g.: codes, names, etc	☺		Functionality exists, but view is unusable Also no synchronisation between local and server ontology. No link between ontologies either.
Search for related ontological resources	For a resource display the values from object properties and datatype properties e.g. “Ostrica” returns “ostrica gigas, ostrica viriginica, etc. and also from commodities “oysters fresh, oysters smoked, oysters canned. In addition it should be possible to see at least the links/titles of underlying data sources linked to such concepts	☺		Does not exist.
Browse Taxonomy	Tree view; It is possible to go from ostrica gigas up to ostrica,	☺		Rubber-band view but is unusable (vibrates). No tree view.
Query Composition	It is possible to build a query based on the ontologies, and that the query choices are a function of the concepts and properties	☺		There are static query screens that are unchangeable and do not modify with any changes to underlying ontologies. This is unacceptable.
Configurable Query Composition	There should be a configuration file/template so an admin could customise the query screen for a given ontology set	☺		Does not exist.
Query for Data Source related to individual	Retrieve a data source (i.e. factsheets) by querying the system with elements from the loaded ontologies	☺		Yes but just returns a link, and just for a Lucene index. There is no visible relation between the result and the ontological concepts to which it is purportedly related.
Visualize query results	Request info on abundance level of aquatic resources for which exploitation status is “fully exploited”, get list of such aquatic	☺		Does not exist

Use Case	Comment	Priority		First Prototype
		High	Low	
	resources together with abundance level. Also get links to data source (i.e. factsheets)			
Visualize Data Source related to individual		☺		Currently just an unclickable link, not even the title.
Refine query	Means it is possible to amend elements from last query	☺		☺
View ontological resource annotation	Annotations already attached to ontology	☺		☺
View Data Source Annotation	Metadata for a factsheet, you would see it along with the URL		☺	Does not exist. Would be nice if not too complicated.

The prioritization of most mentioned use-cases is “high”, representing their core value, and refers to a ranking system we started since [NeOn D7.5.1], and maintained for [NeOn D7.6.1] and [NeOn D7.7.1] A number of other use cases were dropped from the expected functionality, e.g. features representing user management and collateral features such as sending search results as email or maintaining a set of favourite searches. These features were dropped in favour of more knowledge related functionality which demonstrates the power of ontologies for the fishery domain.

2.3 Sources of Information

Every day work of researchers and policy makers on fisheries, and stocks in particular, is based on a number of specialized sources of information (a comprehensive list of which was gathered and analyzed in deliverable D7.1.2 [NeOn D7.1.2]) which always include numeric data (i.e., statistical data, aka time series on aspects such as fish catch, production, fisheries commodities import/export etc.) and textual data such as reports, fact sheets and white papers. For this reason, two data sets representative of these two different sources of information were chosen as a basis for the FSDAS: the set of ontologies reengineered from the reference tables used to index time series (Section 2.3.1) and the fact sheets on fisheries produced by FIES, the “Fisheries and Aquatic Information and Statistics Service” within the Fisheries Department FI (Section 2.3.2).

This set of ontologies is the cornerstone of FSDAS building the network of ontologies that allows extracting, analyzing, and aggregating information for a fish stock assessment system. The conceptualizations are used to represent actual fisheries data, such as the geographical regions, the habitats actual fish stocks live in and are the vessel and gear types that harvest them. FSDAS makes this information available, which is originally stored in a large number of XML files – the so called FIGIS fact sheets (cf. Section 2.3.2).

2.3.1 Reference data for FAO time series on fisheries

FIES collates time series concerning various aspects from fisheries. Data is provided by member countries and is then stored (and published) according to classification systems used to reference each piece of statistical data.

A time series is a sequence of statistical observations which are ordered in time and/or space. FIES collects observations about captures, aquaculture production, catches, fleets, trade of commodities, and consumption [FISTAT]. Any piece of statistical data is referenced by the following dimensions: time (in years), space (land and/or water areas), and the variable representing the observed object (e.g., biological species). For example, we can have statistical data about the catch of a given species in a given water area over a certain time span. In the case of statistics concerning trade, also the "trade flow" (import/export) is included. All statistics collected by FIES are available on the web and accessible by means of an online query panel.[1]

Reference data (usually referred to as RTMS) has been reengineered into ontologies in order to exploit the possibilities of semantic technologies. This work, carried on in T7.2, resulted in six OWL ontologies published online[2]; also, both the resource on which the ontologies were based and the ontologies were described and discussed in detail in deliverable D7.2.2 [NeOn D7.2.2]. At the time of writing, those ontologies are being revised and included into a network of ontologies (cf. D7.2.3, forthcoming).

The ontologies included in D7.2.2 are about: land areas, FAO divisions of water areas used for fishing areas, biological entities, fisheries commodities, vessel type and size, gear types.

All ontologies are in OWL (serialized as RDF/XML) and they all adopt a consistent modelling style: a limited number of classes and large amount of instances (in other words, the actual pieces of information are modelled as instances, ABox); names are available in English, French, Spanish (although English is usually the most represented) and often more than one name is available in each language (cf. the case of countries, for which we have short and long official names); instance names are given by a combination of their "meta code" and ID (for details on this, see description of the database contained in D7.2.2).

The ontology on land areas contains information about self-governing countries and some groups they form, namely (geographic or economic). This information is important since most fisheries statistics are reported by individual or groups of countries. The codes contained in this ontology are the ISO-2 and ISO-3, the UNDP code and the M49 code.

The ontology on fishing areas is about the FAO division of water bodies used for statistical reporting. This coding system divides the world into a number of main areas, which are further divided into subareas, divisions, subdivisions. The division of water areas forms a strict and complete hierarchy.

The ontology on biological entities contains the taxonomic classification of species of importance to the work of FAO in fisheries. The original list is organized and maintained in the Aquatic Science and Fisheries Information System (ASFIS) [ASFIS] and currently includes nearly 11.000 species items related to Fisheries and Aquaculture.

The ontology on vessel types and size organizes the information necessary to assess fleet capacity and main characteristics of vessels, such as their size or length. The ontology includes information from classifications used for vessel size, the Gross Register Tonnage (GRT), as defined by the Oslo Convention (1947); and the Gross Tonnage (GT) as defined by the 1969 London Convention.

[1] http://www.fao.org/fishery/statistics/en

[2] http://www.fao.org/aims/aos/fi/neon.jsp

Finally, the ontology on gear types is about the International Standard Statistical Classification of Fishing Gear (ISSCFG) [ISSCFG] classification of gear types. The type of gear installed on a vessel determines the type of fish that it can catch and therefore it is often used to determine the fleet power. Although this classification was initially designed to improve the compilation of harmonized catch and effort data and in fish stock assessment exercises, it has also been found to be very useful for fisheries technology, fishermen training and for the preparation of specialized catalogues on artisan and industrial fishing methods.

2.3.2 Fact Sheets on Fisheries

FAO also generates a number of fact sheets about various subjects in fisheries. The fact sheets contents[3] is generated by experts and published electronically as XML documents according to a comprehensive XML schema [FSschema]. In this way, FAO makes available a large amount of information about fisheries, aquaculture and related subjects, including fishing techniques, fishing areas, fisheries and aquaculture country profiles. Fact sheets are grouped by *domains* (e.g., Cultured species, Fishing equipment, Fishery, Gear type), each corresponding to an element under the root FIGISdoc, the root of any fact sheet (XML document). Domains are fully specified by means of nested elements. Each element includes a description meant for human use.

The schema makes use of existing standard element sets such as Dublin Core [DC], Extended Dublin Core [EDC], AGMES [AGMES] and AIDA [AIDA]. It also incorporates wherever possible existing classification schemes (such as ISO standards for countries, currencies, languages, and other fisheries-related international classification schemes) most of which are stored in the RT.

It is important to note that the schema was conceived as a means for editors to create structured documentation, and as such was not created based on a relational or ontological model, but was rather organized following hierarchical document formatting conventions. A dictionary of the elements used in the schema is available online [FSdic].

[3] The actual factsheets published in http://www.fao.org/fishery/factsheets/en and http://www.fao.org/fishery/rfb/search/en are usually the result of a complex procedure to make the content provided by the experts available in a format suitable for publication, after integration of information sotred in other systems (e.g. geographical maps).

3 Ontologies

The FSDAS application is conceptually linked together by a number of networked ontologies. An application-ontology builds a hub that reaches out to the different taxonomic ontologies (presented in Section 2.3) and thus creates a network of ontologies for different aspects of the fisheries domain. The application ontology also interacts with the available basic factual information about different observations represented in the form of XML fact sheets and links them to the taxonomic information from the RTMS ontologies.

The original data sources are only very loosely linked to each other via the use of codes. E.g. the fact sheets about aquatic resources refer to water areas or species via alpha-numeric codes. Within FAO many different coding schemes exist which must be taken into account when integrating many fact sheets and presenting them in a unified way to users or to make them searchable via a single search interface.

It takes a well-trained user (or a suitable application) to determine the semantics of the different codes or even to locate descriptions for entitles, which are represented by these codes. This task is now enabled in FSDAS via the RTMS-based ontologies.

In addition to linking the basic data to background information these RTMS-based ontologies also capture all taxonomic knowledge relevant for FSDAS. The main contribution of these ontologies is the translation of the coding schemes and very implicit taxonomic relationships between entities from the RTMS database into ontological individuals (that have an identity) with properties (like taxonomic codes in different coding schemes) and taxonomic or hierarchical relationships to other individuals.

The following sections will present the application ontology and how it was conceived.

3.1 Competency Questions

In order to assess the scope of relevant knowledge that should be accessible through the system, we organized a workshop to learn about the domain knowledge and to structure it. During this workshop we collected a number of competency questions from members of the Fisheries department from FAO. These questions were the main input for conceptualizing the classes and relationships for the application ontology.

1. Give me the species having a "demersal" habitat in water area "24".
2. Give me the synonyms and local names for species "Ostrica gigas".
3. Give me the species with which species "Ostrica gigas" can be confused.
4. Give me the species found below 200 metres for water area 24.
5. Give me the species that eat "shrimp".
6. Give me the species eaten by "seals".
7. Give me the species caught using "bottom gillnets".
8. In which water areas are "bottom gillnets" used?
9. Give me the species caught using "gillneters".
10. In which water areas are "gillneters" used?
11. Give me the species containing local name "oyster".
12. Give me the species for which conservation status contains "Vulnerable".
13. Give me the Species for water area 24.

14. Give me the Water areas for species "Gadus morhua"
15. Give me the resource observations for the year 2004.
16. Give me the resource observations where the exploitation rate is "Moderate fishing mortality".
17. Give me the resource observations where the state is "fully exploited".
18. Give me the resource observations where the abundance level is "Low abundance"
19. Give me the resource observations where the jurisdictional distribution is "Highly migratory".
20. Give me the resource observations where the zone is "Tropical"
21. Give me the resource observations where the vertical distribution is "Pelagic".
22. Give me the gears targeting species "tuna".
23. Give me the gears which incidentally catch species "dolphins".
24. Give me the vessel types that use "trawls"

This set of questions introduces the vocabulary for building the application ontology for FSDAS. They nicely relate the user needs with existing facts from the fact sheets and allow us to build the FSDAS ontology and populate it from the fact sheets. The questions were created with some background knowledge about the available information from the fact sheets and also considered real-world use-cases, i.e. information needs by actual fisheries experts.

From these questions we could immediately extract a couple of classes, like species, water area, resource and resource observation. Some clear relations between these classes were also immediately accessible like “species eating other species”, or “gears targeting species”. The third kind of knowledge gathered from the competency questions are the properties of the classes, such as “water areas have IDs”, “resource observations have exploitation rate and state”.

After some more investigation and interviews we identified for some of the classes, that they actually are closed lists, with a pre-defined set of possible values, such as the different horizontal levels of the sea.

The next section describes the FSDAS network of ontologies, developed with pattern-based design methods [NeOn D5.4.2][NeOn D2.5.1] starting from the competency questions. This network has been agreed as the reference ontology for the Application Ontology described in Section 3.3.

3.2 Design of the FSDAS Network of Ontologies

eXtreme Design (XD) [NeOn D5.4.2][NeOn D2.3.2] is a method to design ontologies based on *ontology design patterns* [NeOn D2.5.1][Gangemi&Presutti09], i.e. highly reusable practices for ontology design, and on some principles inspired by eXtreme Programming, such as pair-programming, incremental refinement and unit test-based evaluation (that are used to perform EXD (Evaluation based on XD, [NeOn D2.3.2]), etc.

The ODP portal: http://www.ontologydesignpatterns.org [DagaEtAl08] is a repository of Ontology Design Patterns, currently populated with Content Patterns [Presutti&Gangemi08], which are small reusable ontologies that encode a good solution to one or more

competency questions [Gruninger&Uschold95] i.e. typical questions that an expert might want to ask to a knowledge base including entities and facts from its domain of expertise.

We have used XD and EXD, together with some content patterns taken from ODP, in order to design the OWL modules to be reused in the FSDAS Application Ontology (see next section).

FSDAS prototype intends to visualize fishery information from multiple FAO resources that are queried by using terminology extracted from FIGIS XML schemas, RTMS tables, and (in the future) ASFA Thesaurus [NeOn D7.2.3]. In order to reach that goal, we need to make the data interoperable, and the related activities are summarized here:

(a) conversion of XML schemas into "schema-based ontologies"

(b) design of a "network of ontology modules" from FAO competency questions (this section)

(c) deriving an Application Ontology from the network of modules (next section)

(d) mapping the elements of the Application Ontology to the elements of the schema-based ontologies (this deliverable)

(e) querying the information resources through the Application Ontology

(f) visualizing a query interface and the data eventually obtained.

Therefore, the network of ontologies mentioned here is the result of (b).

We started from the 24 competency questions listed in Section 3.1 (and in their original form in the ODP wiki[4]), and we analyzed them in order to find possible commonalities or inclusions, and to produce actual unit tests to be used in order to evaluate the modules designed based on the competency questions.

In the context of pattern-based design, the competency questions provided by FIGIS are called *scenario questions*, since they contain individual values, and cannot be used as templates for domain competency questions until the individual values are “abstracted” to their class.

For example, the question *Give me the species found below 200 metres for water area 24* should be abstracted to *Give me the species found at a certain bathymetric range for a certain water area*. Abstraction is one of the crucial tasks to be executed during pattern-based design, and is typically performed by close analysis of existing resources (in this case, XML schemas), or with the help of elicitation from experts.

The result of the analysis consisted initially in 15 *leading* scenarios, subsequently abstracted as Leading Competency Questions (LCQ), and of 9 variations of the leading ones (i.e. they are already covered by the leading ones).

LCQs have been used by a team from the STLab of CNR, including Aldo Gangemi, Eva Blomqvist, and Alessandro Adamou (and additional discussions with Valentina Presutti), in order to produce a network of 14 ontologies in the form of Content Patterns, which were published on the ODP portal[5] for public evaluation (where fishery experts can review them, but everyone registered to the site can review them as well).

[4] http://ontologydesignpatterns.org/wiki/Community:FSDAS_Scenario

[5] http://ontologydesignpatterns.org/wiki/Community:FSDAS_Scenario_2

The design process of the team has been collaboratively performed by using the ODP portal with *modeling issues*, *evaluation wikiflow*, and *pattern proposal* functionalities [DagaEtAl08].

Argumentation sessions by using Cicero Wiki [NeOn D2.3.1] have also been opened in order to discuss several modeling issues. These issues included for example:

- Reusing existing patterns in ODP. For example, we reused the content pattern situation.owl[6]
- Deciding if preliminary FSDAS content patterns can be used in order to compose other, more complex ones. For example, we reused aquaticresources.owl [7] in most FSDAS patterns because it contains the basic modeling solutions to relate *aquatic species* to *aquatic resources* and *water areas*, which can be considered the *core* entities for fishery (Fig. 1)
- Deciding on the naming patterns to be applied with respect to formal semantic issues, e.g. should we call a class *Gear* of *GearType*? We analyzed the scenarios, and there the domain of interpretation is clear: only gear *types* are considered
- Deciding on logical/transformation patterns to apply, e.g. should an observation be considered a time-indexed relation to be reified by using the *n-ary relation* pattern? In this case, we decided to generalize over several observation-based LCQs, since *prima facie* they seemed to require time indexing, and specialized the pattern observation.owl[8] for all of them as the pattern aquaticresourceobservation.owl[9].

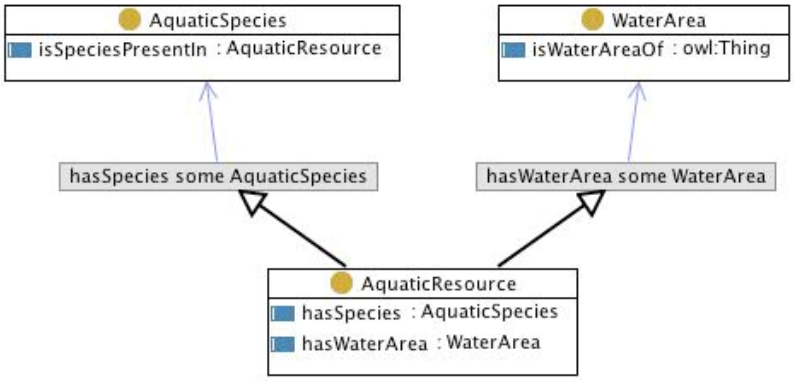


Figure 1: The `aquaticresources.owl` content pattern. Bold arrows indicate rdfs:subClassOf; thin arrows indicate associations via rdfs:domain or rdfs:range.

After publication on ODP, the FSDAS content patterns have been submitted to Aureliano Gentile, the FAO chief designer of FIGIS information systems, and elicitator of fishery expertise. His evaluation of the initial network of ontologies evidenced some flaws in our team's decision, mainly because the design was solely based on scenario questions, without a domain expert available to resolve fishery conceptualization issues at argumentation time. The three main arguments can be summarized as follows:

[6] http://ontologydesignpatterns.org/cp/owl/situation.owl

[7] http://ontologydesignpatterns.org/cp/owl/fsdas/aquaticresources.owl

[8] http://ontologydesignpatterns.org/cp/owl/observation.owl

[9] http://ontologydesignpatterns.org/cp/owl/fsdas/aquaticresourceobservation.owl

1. *Observations*. Not all observations seem to be time-indexed for fishery experts: some of them, like *exploitation rate* and *abundance level*, are clearly temporalized, but others are not, like *vertical distance* and *jurisdictional distribution*. Fishery experts seem to converge on "average" or typical distances or distributions for aquatic species, and assume them by default. However, the expert admitted that some values could change as far as actual aquatic resources are concerned, but this is a matter of empirical findings.
2. *Local bathymetric ranges*. In the initial design, it seemed conservatively safe to assume that bathymetric ranges of species/resources can change for different water area distributions; therefore the situation.owl pattern was reused, in order to implement a vocabulary that enables the n-ary relation logical pattern. However, similarly to argument (1), fishery experts assume a default bathymetric range for their practical work, hence there is no need any more for an n-ary pattern.
3. *Compositional properties*. In the initial design, our team has applied XD straightforwardly, considering competency questions as the closest indicator of fishery expert conceptualization. However, the experiment proved that a more sophisticated procedure is needed when analyzing competency questions. For example, given the scenario: *in which water areas are "bottom gillnets" used?* and the abstracted competency question: *in which water areas is a gear type used?*, the first pattern was designed by creating an object property `isUsedInWaterArea`, with `rdfs:domain` *GearType*, and `rdfs:range` *WaterArea*. However, it seems that hardly any expert would conceptualize a relation like this directly, but only as the result of a *composition* chained through aquatic species: if a gear type is typically used to catch an aquatic species, and the resources containing that species are typically present in a water area, *then* we can say that a gear type is used in that water area.

The evaluation made by the expert provided good rationales for finalizing the (current) version 2 of the FSDAS network of ontologies (Fig. 4).

Concerning argument 1: the pattern developed in the initial phase, based on the `observation.owl` pattern: `aquaticresourceobservation.owl`[9] ,has been refactored into five different ones:

- `resourceexploitationobservation.owl`[10]
- `resourceabundancenobservation.owl`[11]
- `jurisdictionaldistribution.owl`[12]
- `climaticzone.owl`[13]
- `verticaldistance.owl`[14]

An example from resourceexploitationobservation.owl is shown in Fig. 2.

[10] http://ontologydesignpatterns.org/cp/owl/fsdas/resourceexploitationobservation.owl

[11] http://ontologydesignpatterns.org/cp/owl/fsdas/resourceabundancenobservation.owl

[12] http://ontologydesignpatterns.org/cp/owl/fsdas/jurisdictionaldistribution.owl

[13] http://ontologydesignpatterns.org/cp/owl/fsdas/climaticzone.owl

[14] http:// ontologydesignpatterns.org/cp/owl/fsdas/verticaldistance.owl

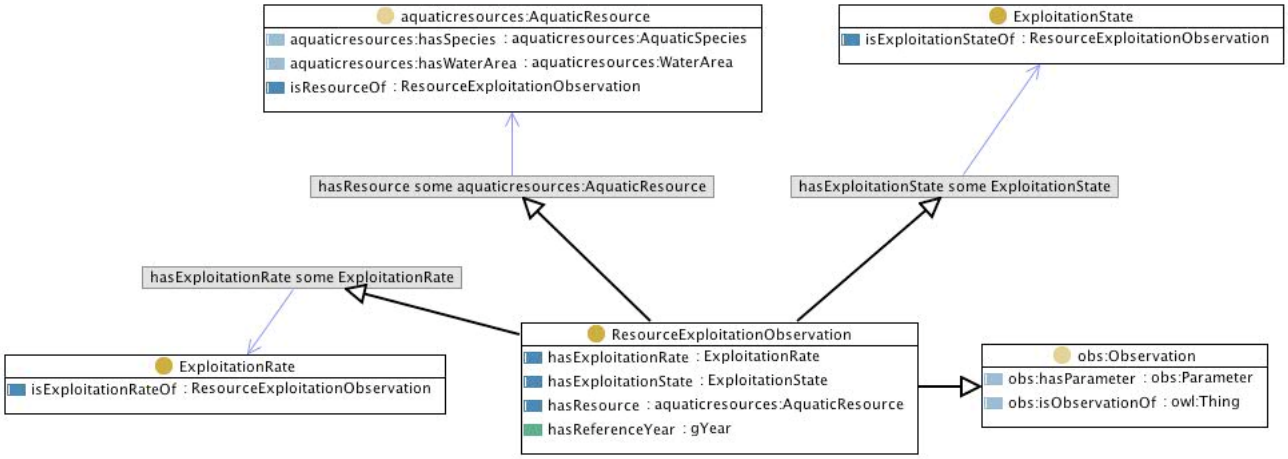


Figure 2: The resourceexploitationobservation.owl content pattern.

Concerning argument 2, we have simply redesigned the pattern `bathymetricrange.owl`.

Concerning argument 3, a different architecture has been designed which facilitates some form of composition, and three patterns have been implemented consequently:

- `gearwaterarea.owl`[15]
- `vesselspecies.owl` [16]
- `vesselwaterarea.owl` [17]

As an example, this is the case with `gearwaterareas.owl` (Fig. 3), where initially we had implemented the relation `isUsedInWaterArea` as a primitive relation between gear types and water areas.

However, as the expert's argument suggests, we need something that is able to represent that *if* a gear type is typically used to catch an aquatic species, *and* the resources containing that species are typically present in a water area, *then* we can say that a gear type is used in that water area.

This must be inferred compositionally, and requires either a more sophisticated logical pattern, like 'property chain', available in OWL2 (but not in OWL1), or a complex reasoning pattern, like 'OWL1 + DL classifier + SPARQL', or 'OWL1 + DL classifier + SWRL rule firing'. For example, with'OWL1 + DL classifier + SPARQL', we can use the following SPARQL:

```
CONSTRUCT {
?x :isUsedInWaterArea ?y . ?y :isSuitableForGearType ?x}
WHERE {
?x gearspecies:targetsSpecies ?z .
?z aquaticresources:isSpeciesPresentIn ?w .
?w aquaticresources:hasWaterArea ?y}
```

The SPARQL code shows that in order to build the query, we need to use vocabulary from

[15] http://ontologydesignpatterns.org/cp/owl/fsdas/gearwaterarea.owl

[16] http://ontologydesignpatterns.org/cp/owl/fsdas/vesselspecies.owl

[17] http://ontologydesignpatterns.org/cp/owl/fsdas/vesselwaterarea.owl

two different patterns, which are therefore “composed” in gearwaterarea.owl:`aquaticresources.owl` and `gearspecies.owl`.

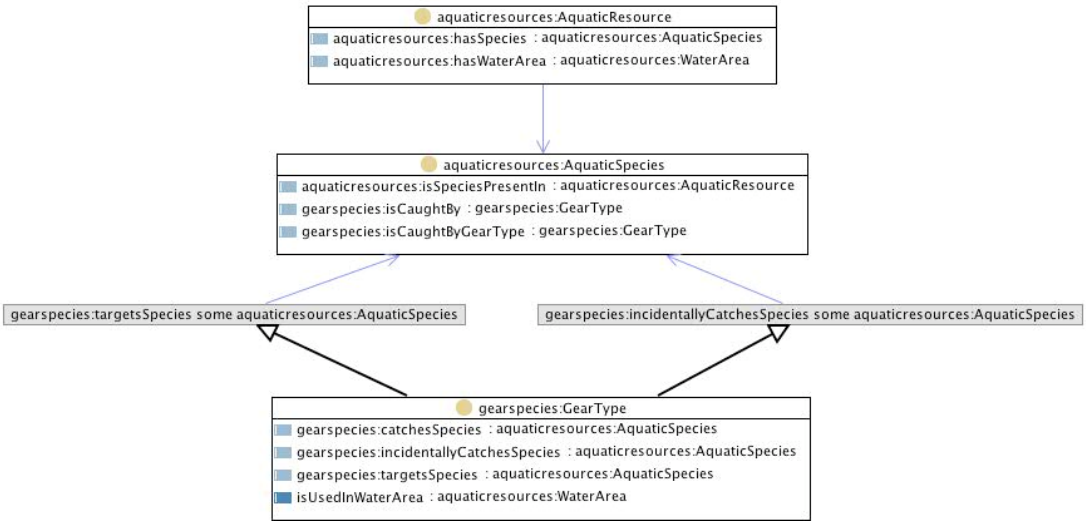


Figure 3: The `gearwaterarea.owl` content pattern.

The FSDAS network of ontologies[18] consists now of 21 ontologies, designed after 17 "Leading" Scenarios and LCQs, and 7 more that are variations of the leading ones (see Footnote 5, where the modeling issues and related solutions are documented in the ODP wiki). 17 of them are fishery content patterns, one contains the closure of all owl:imports axioms (cf. Footnote 18). one is the pattern annotation schema[19], and two are patterns reused from ODP (cf. Footnotes 6 and 8).

[18] http://ontologydesignpatterns.org/cp/owl/fsdas/fsdasnetwork.owl

[19] http://ontologydesignpatterns.org/schemas/cpannotationschema.owl

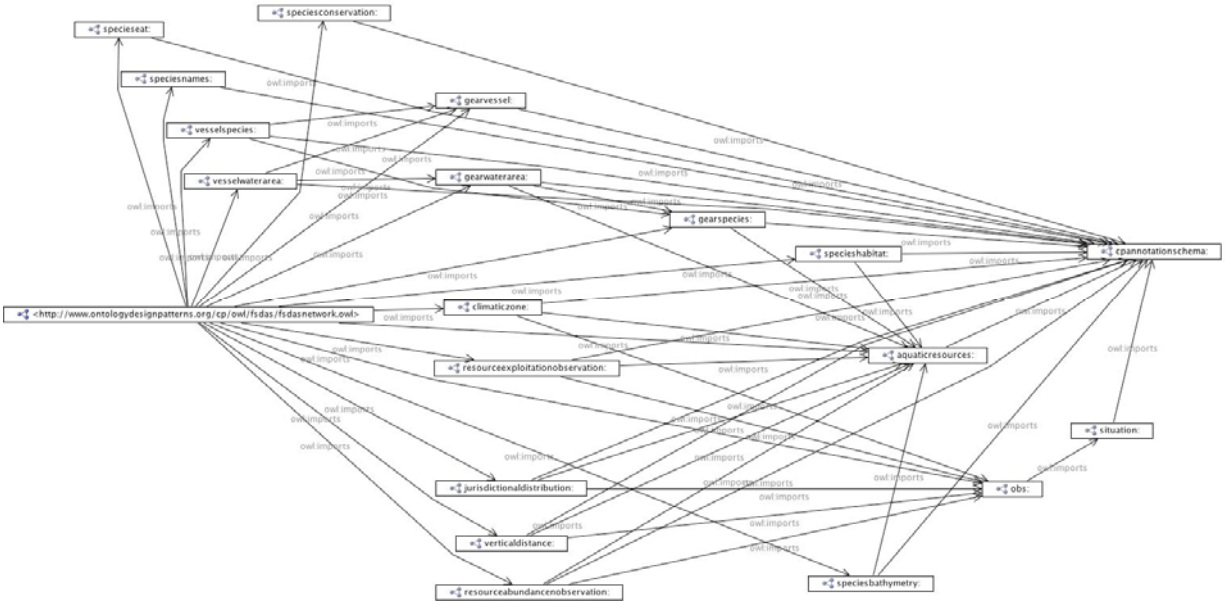


Figure 4: The FSDAS network of ontologies: fsdasnetwork.owl imports all modules.

Each pattern is richly annotated by using the same annotation schema as the one employed for content patterns, and can be downloaded from the ODP portal ('Proposed content patterns' section), or directly from the links presented above.

For most classes in the patterns, in the annotation axioms for `rdfs:comment`, we have tried to figure out what is the closest mapping to the schema-based ontologies, but this has been done with reference to a conversion made originally by CNR[20], because the translation provided by the SAG tool[21], although rather complete (except for enumerations and comments), generates very long and hardly readable name, which makes the mapping activity quite difficult. Our annotations are anyway usable for task (c).

3.3 The FSDAS Ontology

The knowledge base used for the FSDAS application is separated into two ontologies. The actual FSDAS v2.0 application ontology and the *facts* ontology containing the ABox axioms extracted from the fact sheets. The facts ontology imports the axioms from the FSDAS ontology via the OWL import mechanism.

The application ontology presented here differs from the pattern network presented above. It uses only a subset of the vocabulary defined in the ontologies because the application is (currently) restricted to a subset of factsheets. In the application ontology the classes and properties from different ontologies from the pattern network are merged into a single ontology to simplify the access from within the application.

[20] http://ontologydesignpatterns.org//ont/fao/figis/speciesNTK.owl

[21] An example of a translation http://ontologydesignpatterns.org//ont/fao/figis/fintk.owl

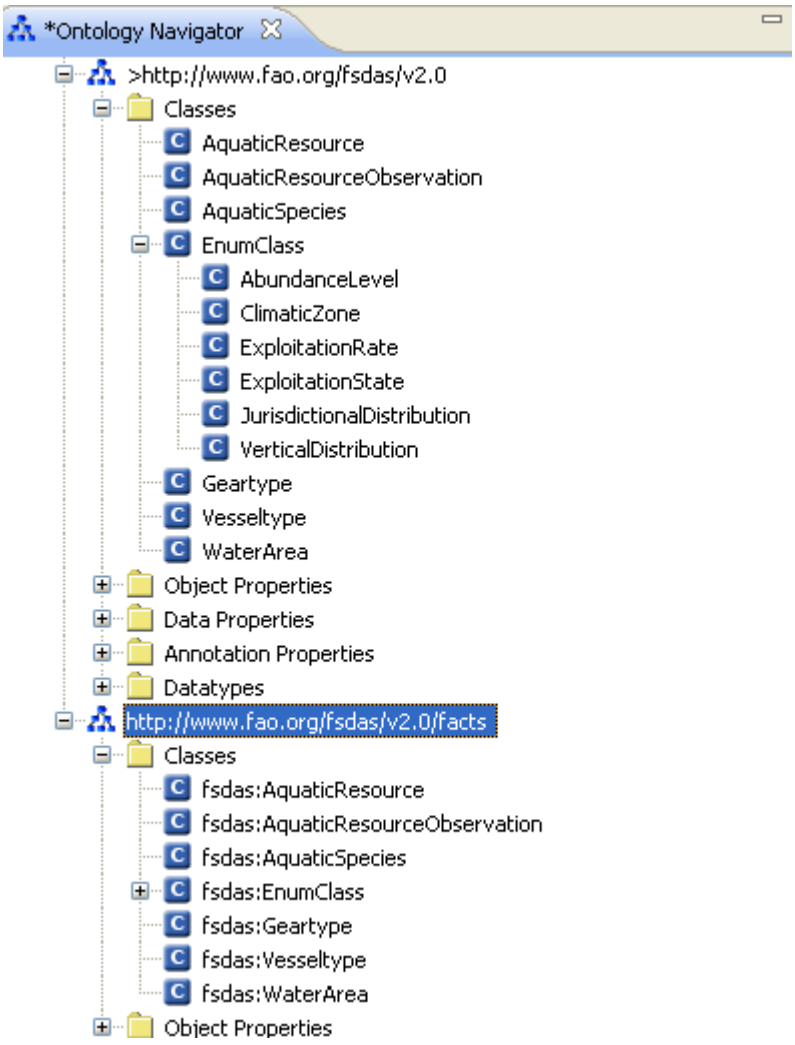


Figure 5: Overview of the taxonomy of classes of the FSDAS ontology

The main classes of this ontology are

- AquaticResource
- AquaticResourceObservation
- AquaticSpecies
- WaterArea
- and the enumeration classes which represent sets of individuals with fixed semantics

AquaticResources represent static knowledge about resources, such as the species and water areas they refer to. Actually an *AquaticResource* is defined by its species and water areas.

AquaticResourceObservations are in some sense specific *AquaticResources*. They hold information about one resource in one specific reference year. The information about the resource includes the abundance level, the exploitation state and the exploitation rate.

AquaticSpecies are the fact sheet equivalents to the the RTMS species class. This class holds information about the different species, such as similar (looking) species, vessels and gears used to catch them, or prey relationships between different species. In this way it complements the taxonomic knowledge from the RTMS ontology with non-taxonomic relationship between species.

The subclasses of *EnumClass* represent individuals which are used in a lot of places and have a well defined semantics. The population mechanism translates the textual strings found in the fact sheets into individuals with a proper ID to represent the meaning of a specific property.

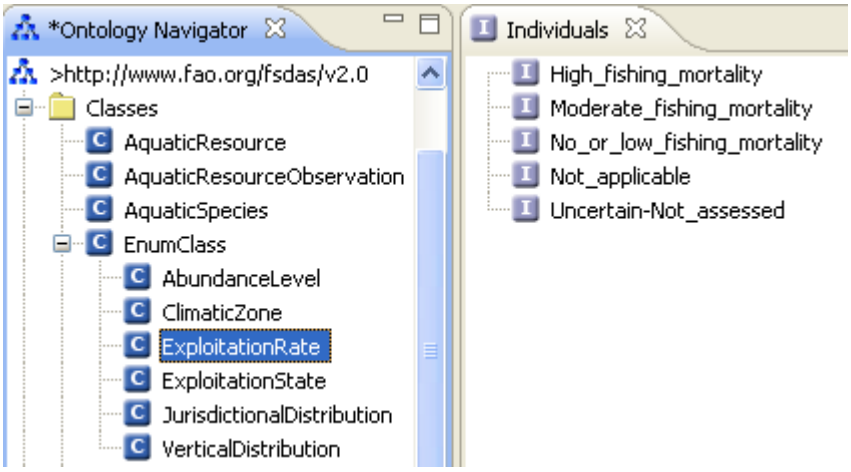


Figure 6: Individuals of the class ExploitationRate

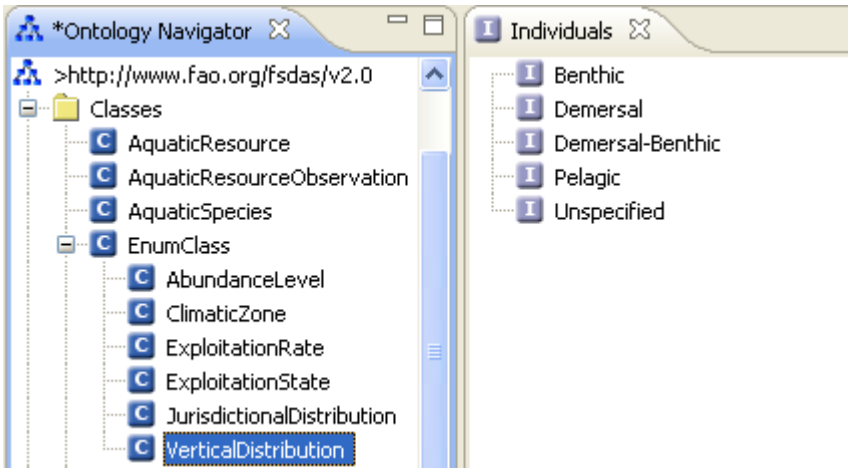


Figure 7: Individuals of the class VerticalDistribution

The other classes in the FSDAS ontology represent fact sheets with less rich content.

WaterArea factsheets hold only very little information and thus the ontological class is also quite simple. Water areas are identified by their FAO area code and have a human understandable name and descriptive text.

Vesseltype represents vehicles on the sea that are use to catch fish. Vessels use special gears.

Geartype represent the equipment for fishing, e.g. different kinds of nets. The factsheets for gears contain information about the species which are targeted by the gear and also about the incidental *bycatch*.

4 Architecture and Design

4.1 Overall Architecture

The most striking design decision for the second prototype of FSDAS is probably the migration to a web-based frontend. Initially, when the original requirements for FSDAS have been formulated about 2 years ago, this requirement of an installation-free and browser-based application was already present. Version 1 of FSDAS, though, was based on the Eclipse and NeOn Toolkit platform to optimize component reuse as much as possible. Unfortunately, these benefits only helped developers but not users. Users were asking for a much more streamlined FSDAS application that does not contain any functionality that is not needed (and *sneaked* into V1 via the used framework).

The overall architecture of FSDAS v2 (cf. Figure 8) resembles the traditional three-tier organisation of functionality which is distributed to a client and a server machine.

1. On the top layer we have the user interface which is executed in the end-users web browser.
2. The second layer provides the business logic and feeds the client with layout information and the needed data for populating the GUI.
3. The third layer implements the traditional data storage layer. In the case of FSDAS, this layer is implemented using semantic technology and wraps ontologies, basic facts and reasoning capabilities which are exposed via a simple API to the middle logic tier.

The presentation tier is implemented as an AJAX application, which communicates with the logic tier via the HTTP exchange mechanism. This presentation tier provides the overall user interface of the application in a web browser, using the ZK[22] technology, an open-source Ajax framework, which enables us to create rich web applications which create a user experience very similar to the desktop and thus familiar to many users.

The logic tier and the data tier reside on the same server machine and also share a common Java virtual machine. The communication between both layers is realized in pure Java making use of the KAON2 API in particular Java classes for ontologies, classes, properties and OWL axioms. The API also exposes a SPARQL interface to query the ABox and thus allowing retrieving facts that have been gathered from different XML fact sheets into a unified model that adheres to the vocabulary defined in the FSDAS ontology and to the terminology introduced by the set of RTMS-based ontologies.

[22] http://www.zkoss.org

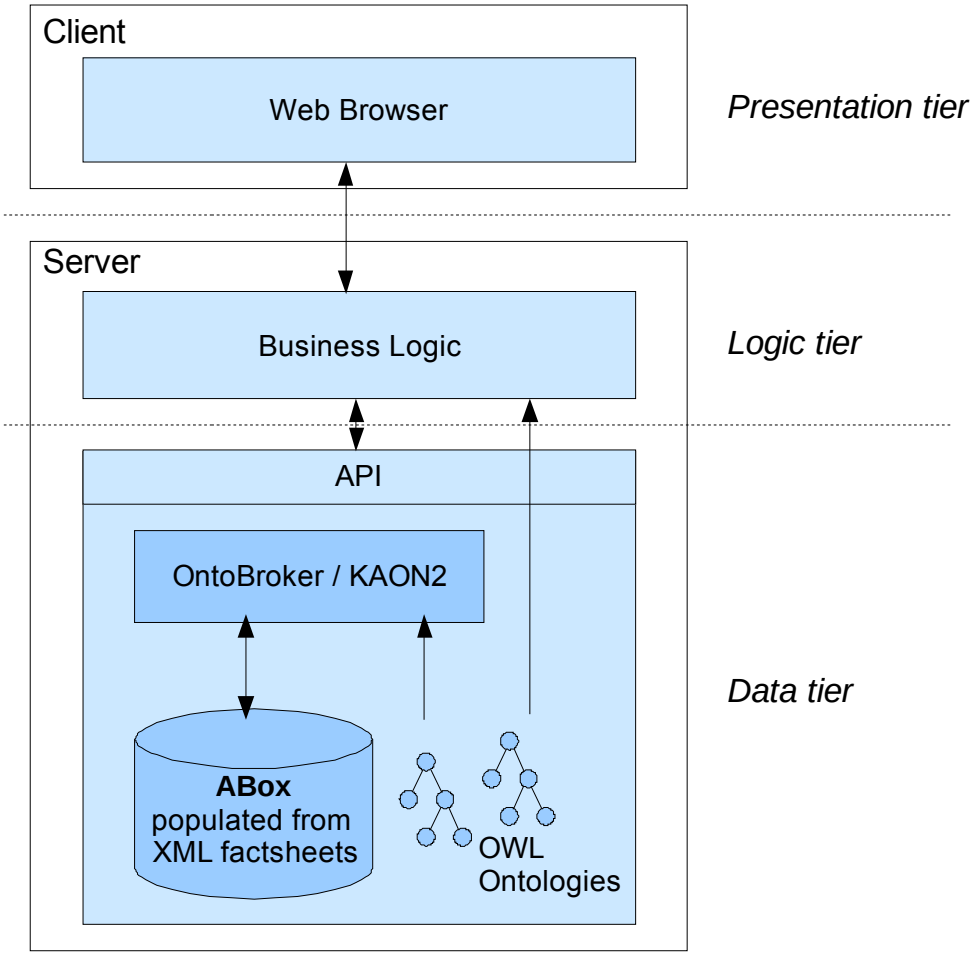


Figure 8: Overview of the FSDAS v2 architecture

4.2 Population of the Knowledge Base

The process of populating the knowledge base with the XML factsheet content is similar to the human-oriented publication process for creating HTML files for presentation on the web. The current process within the fishery department is presented in Figure 9. It takes some manually created XML files and enhances their content with additional information from databases and thesauri to create enriched (output) XML factsheets. These factsheets are the input for the actual publishing process on the web which presents XSLT-generated HTML pages to the human reader.

The process of extracting semantic information from the XML factsheets is very similar to the web-publishing process. It also starts from the enriched (output) XML factsheets and produces RDF data via an XSL style sheet (cf. Figure 10). Since not all needed information can be extracted in the proper form with the XSLT processing an additional completion step was introduced which makes sure that a proper RDF graph is created from the XML factsheets. In particular it is necessary to create RDF resources for the different entities represented in the factsheets. Since the factsheets define the relationship between different entities, e.g. an aquatic resource and a species via a key/foreign key relationship or even via matching (non key) attribute values, e.g. the scientific name the associations are first extracted literally with the XSLT style sheet and in a second step they are consolidated and transformed into real associations between objects (this especially requires the identification

and creation of the correct resource identifiers, according to a well defined naming scheme, based on the RTMS taxonomy of codes).

This process yields one RDF fact file for each XML factsheet. The final step of the KB population process is simply the import of the set of RDF files into a single Knowledge Base (cf. Figure 11). This KB is managed by OntoBroker/KAON2 and is connected to the FSDAS ontology and to the terminology defined in the RTMS-based ontologies. As can be seen in the architecture overview earlier this KB is exposed to the FSDAS application (logic and presentation tier) via an API providing a SPARQL interface as well as rich, convenient KAON2 objects.

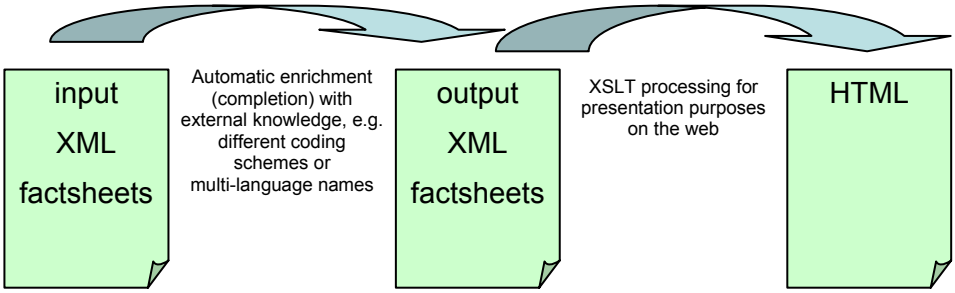


Figure 9: Publishing process of factsheet data

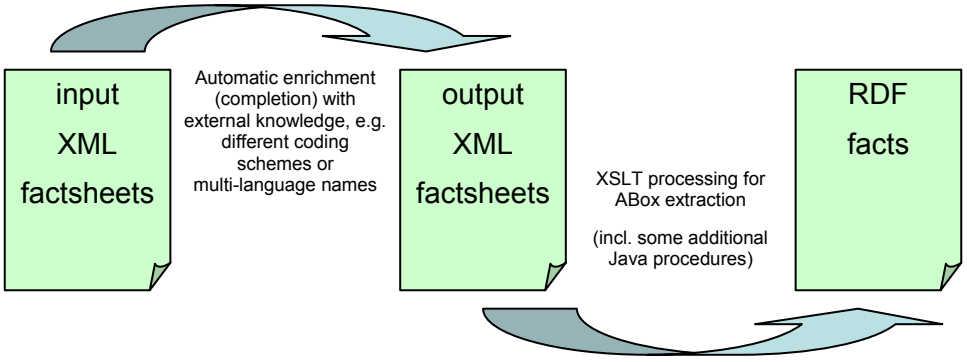


Figure 10: Extraction process for factsheet data into RDF facts

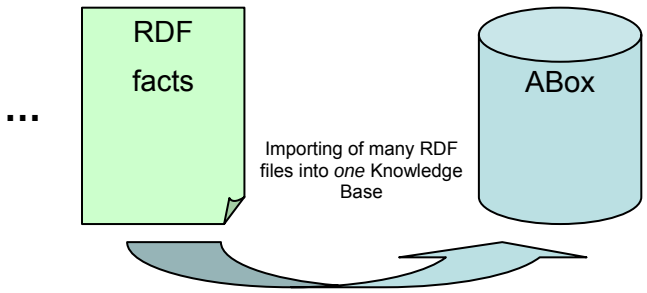


Figure 11: Merging of many RDF files into a single knowledge base

4.3 Server Layer

4.3.1 Back-end-related Server

The back-end consists of the OntoBroker inference server that manages the knowledge base formed of the extracted fact sheets and the RTMS ontologies. OntoBroker is a kind of deductive data base environment. This means that OntoBroker manages not only pure data by storing it and providing mechanisms for retrieving it; additionally OntoBroker also supports rules that describe how implicit data can be derived from already stored facts by inferencing over a given set of facts and rules.

The back-end component of FSDAS provides to the upper tiers of the architecture interfaces to (i) manage the knowledge base and esp. query it with query languages such as F-logic and SPARQL and (ii) the management and access of the RTMS OWL ontologies via the KAON2 API [KAON2].

All implementation aspects and all configuration details are hidden behind an easy-to-use API that encapsulates the access to the data layer and abstracts away from the underlying OntoBroker technology. The lower part of the architecture overview in Figure 8 shows the layout of the data tier.

Table 2: Operations of the Data tier API

Operation signature	Description
`Ontology getOntology(String ontologyURI)`	Retrieves a KAON2 `Ontology` object that represents the ontology whose URI was provided as parameter.
`Set<String> getOntologyURIs()`	Returns the URIs of all available (RTMS) ontologies.
`String[][] executeQuery(String query)`	Executes a given SPARQL query against the knowledge base and returns the result as a list of tuples.

One main goal in the design process of the data tier API was simplicity and less complexity. The result was an API that provides only three operations and uses the W3C standard query language SPARQL for querying the knowledge base. But it is important to note that this API is extended by the KAON2 API since the data layer API allows to retrieve KAON2 objects. Table 2 gives an overview of the provided operations. A detailed overview of the KAON2 API for ontologies and its contents (axioms and entities) is presented in [KAON2].

4.3.2 Front-end-related Server

This section gives an overview of the FSDAS server-side structure. In the three-tier architecture described in the previous section, the server-side implements the business logic (logic tier) of the application.

In the first prototype of the FSDAS, it was defined as a client/server application, where the server layer was basically defined as a Web Service to carry out the data transfers and execution invocation (with communication with the infrastructure: OntoBroker and Lucene

Indexes). For this second prototype, as was explained before, we move the application to a Rich Internet Application based on the ZK framework. ZK is an event-driven, component-based framework to enable rich user interfaces for Web applications. We use the Model-View-Controller pattern for the different processes involved in the application. In the FSDAS application, the View part is implemented as the *zul pages* visualized in the client side and will be explained in Section 4.4 “Presentation Layer”.

In order to better understand the server-side workflow, the following picture will give an overview.

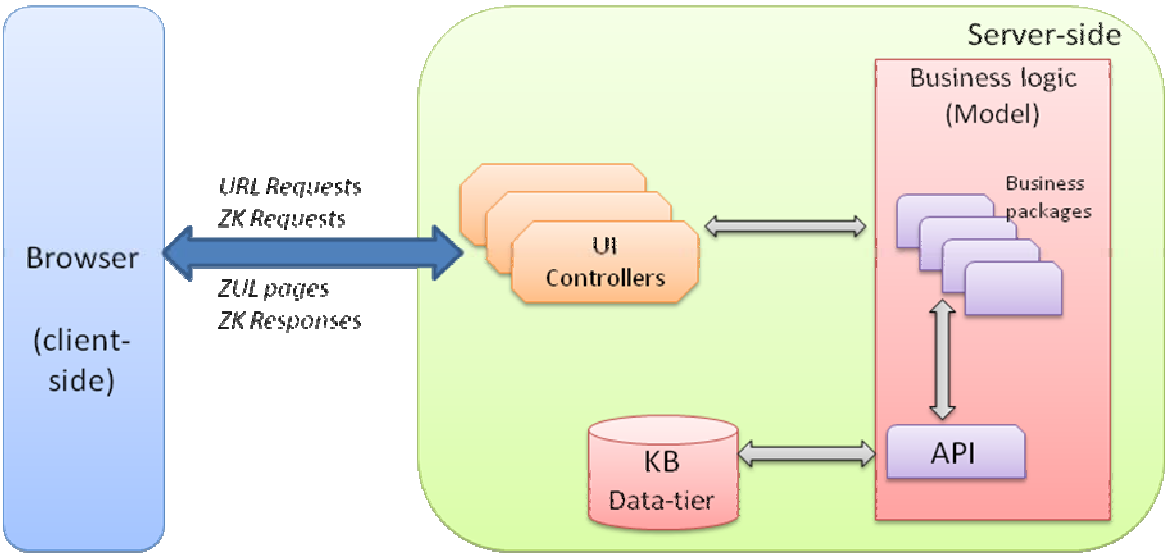


Figure 12: Controllers and Model in the server layer

This section has been divided into two parts, which refer to the main software components in the server side, giving an overview of the connections among them.

Controllers

In the server-layer we distinguish between the *Controllers* and the *Model*. The Controllers are the UI processing components, which allow us to synchronize and orchestrate the user interactions. These controllers are extensions of the main components or widgets involved in the view, like the main window, tree, etc (Java classes from the ZK libraries). These controllers are mainly focused in organize the user interactions and events produced in the view, and separate the business processes involved.

Another task that can be carried out by these controllers is the validation of all input data from the client-side to improve interactivity and to reduce errors caused by invalid data.

The Controllers are located under the org.atosorigin.neon.fsdas.gui.controller package in the server side code. The FsdasController is the class that manages and organizes all the events and user interactions detected in the main window of the web application. According to this, the FsdasController detects the different ZK requests and interacts with the different business processes (Model) designed in the server in order to process the information. When a business process ends, the Controller updates the content of corresponding widget or component, if necessary. These changes are sent to the client through ZK responses.

Other classes involved at this level are in charge of the rendering of the taxonomy trees, entity information component or the results that should be response to the users in the results component. In this case, for the rendering of the big taxonomies of instances involved in the case study, we include an appropriate caching strategy for the business layer (based on HashTables), in order to improve the performance and responsiveness of the web application. With this cache strategy, we avoid unnecessary and duplicated processing.

Model

This section describes the business logic designed for the second FSDAS prototype, including how to interact with the data layer or other functionalities like caching.

The business components are located under the org.atosorigin.neon.fsdas.business package in the server side. In order to minimize the complexity of the development, we separate two areas of concerns in the server: one for the business interaction with the data layer and other for the business workflow.

The model implements specific classes for the main use cases detected in the case study: *search and query against the data*. Depending of the use case and the user input, each method implemented receives different parameters needed for query or search execution and returns a specific chain containing the query results. These classes interact with the query mechanisms provided by the data layer API described before.

4.4 Presentation Layer

Based on the evaluation from the users, mainly focused on the weaknesses in the presentation of the functionality and the general look and feel, the FSDAS has migrated to a web application. For this motivation, the FSDAS second prototype provides a convenient web-based graphical user interface (GUI) for managing and making it easy for the end-users to interact with the prototype and the case study resources. After some interviews and meetings with FAO users, the aspect and organisation of the new Web-based GUI was designed, providing mock-ups that paved the way to the realisation of the GUI for the second FSDAS prototype. We believe that the involvement of FAO users during the design phase will provide an interface better suited to FAO needs, plus future benefits in the evaluation of the prototype. This new GUI is defined as a composition of different widgets. The next section describes the new FSDAS GUI developed using the ZK technology and the user input.

As we explained before, the web-based GUI developed in the prototype is part of the View in the MVC pattern. The design, navigation and look and feel of the FSDAS prototype are supported by ZK technology. ZK is an event-driven, component-based framework to enable rich user interfaces for Web applications. ZK includes an AJAX-based event-driven engine, a rich set of XUL and XHTML components and a markup language called ZUML (ZK User Interface Markup Language).

With ZK, we can present the FSDAS prototype in feature-rich XUL and XHTML components and manipulate them upon events triggered by user activities, similar to what is done in desktop applications. Unlike most other AJAX frameworks, as far as ZK is concerned, AJAX is a behind-the-scene technology. The synchronization of component content and the pipelining of events are done automatically by the ZK engine.

In the following paragraphs we discuss the structure of the GUI's main window and explain the components of the main window, with their functions, and show which of those components are visible in the window when you first invoke the GUI.

From the GUI, the end-user has access to the functionalities expected from the use case scenarios of the FSDAS prototype:

- Search Functionalities: the end-user can search for related ontological resources (species, family, resources, water areas…) or for ontology content as data value. The results are obtained from the data tier through the business logic in the server side.
- Browse Taxonomy: the GUI includes a taxonomy browser of the main ontologies involved in the case study. The instance-level taxonomies are extracted from the ontology models in the server side (at controller level).
- Query Composition: Allow the users to build a query based on the ontologies dynamically. The query component of the GUI is constructed based on a XML template which describes the different possibilities of queries for the user. This template could be used by an admin to customize the query component GUI for a given ontology set.

The FSDAS web prototype can be accessed via the following URL:

http://212.170.156.131:10000/FSDAS_web/fsdas.zul.

4.4.1 Overview of the GUI structure

The FSDAS GUI is divided in the following sections:

Figure 13: FSDAS web application GUI sections

- Taxonomy Widget: Located in the left hand side of the UI you find the taxonomy tree (i.e. species taxonomy, water area taxonomy...) extracted from the FAO ontologies. The user can select which taxonomy should be visualized in the unique taxonomy panel. When the end-user selects one item of the tree, the Entity Information Widget updates its content.

- Entity Information Widget: in the left-bottom side of the UI the widget displays the details of the item selected in the taxonomy widget. The details are extracted from the knowledge base, including object properties, attributes, and annotations.
- Search Widget: is a textbox located at the top-right of the UI. The user can input free text in order to search in the data layer ontological resources for matching entities.
- Search Result Widget: Grid result panel which shows the results of the search widget. It is located in the right of the UI.
- The Query Composition Widget is a panel where users can formulate queries against the knowledge base. It displays a form containing different propererties defined for classes in the ontologies. This form is generated dynamically based on an XML template where the possible aspects that could be queried for are defined. When the user completes his input in the form, the server side transforms this input into a formal query , which gets executed against the knowledge base using the methods provided by the data layer API.
- Query Results Widget: Situated at the centre-bottom of the UI, this component displays a grid with the results returned from the composed query. Currently, this table will at least show the resources name of the retrieved results and actual links to them for further investigation by the user (A click on the link will open a human readable version of the resource). Depending on the query posed additional properties can be shown in the table as well.

A real view of the GUI at the presentation layer is presented in the next figure.

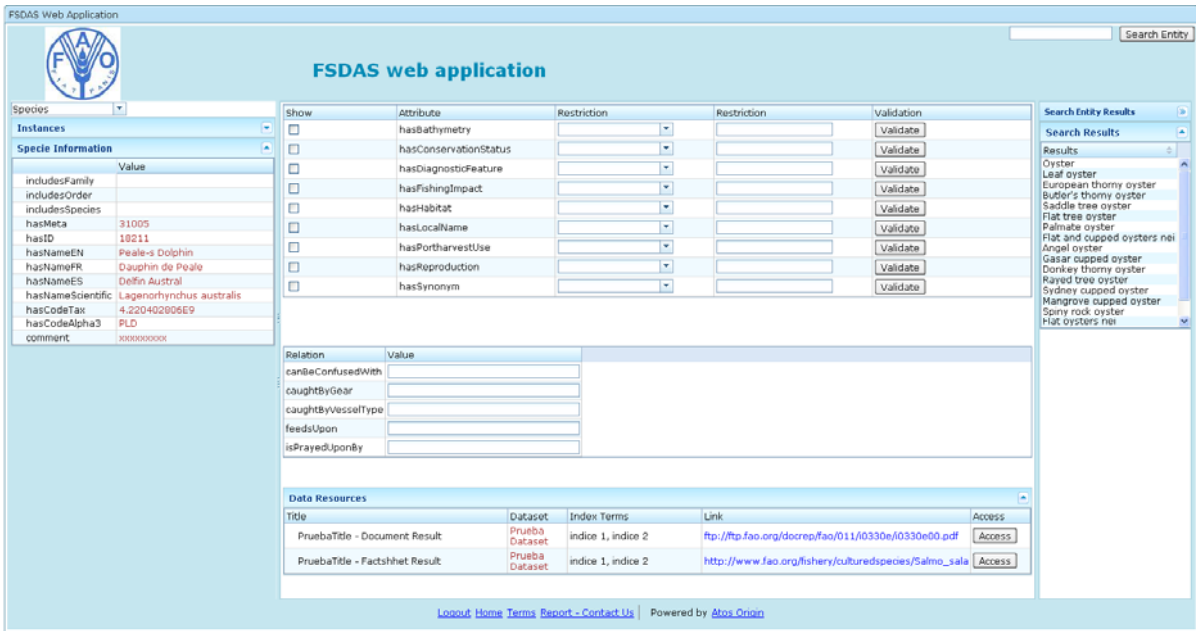


Figure 14: FSDAS Web Application main UI

More details about the GUI widgets and how to use the FSDAS application can be found in the appendix in Section 8.

This implementation of the GUI, based on different widgets included in the main window of the user interface, allow that in future developments of the prototype, more widgets may be

added with different purposes providing more functionalities to the end-users. Moreover, the use of widgets has the advantage that the GUI design can be modified and adapted to the users requirements in an easier way.

5 Conclusion

The development work for FSDAS v2 has been entirely based on the recommendations collected at the time of producing the D7.7.1 (Evaluation of the FADAS first prototype and recommendations to research). By then, suggestions covering functional and non-functional aspects where given as feedback, both, from final user evaluation and technical analysis of the prototype.

The renovated deployment of the client side as a web application, provides a lot more flexibility to the prototype, and brings back a level of higher compliance with the original requirements stated in D7.1.1.

Although the resources available forced the team to scale down the expectations on the available functionalities for this prototype, the ones which have been included have a robust implementation and altogether provide a perception of a solid software platform.

A knowledge base built over the facts extracted from multiple data sources, diverse also with respect to their persistence format (e.g. unstructured text, XML, DB schema), enables users to experience a comprehensive source of information. The ontological layer offered by the network of RTMS ontologies features inference capabilities to discover non-explicit links between elements of the knowledge base. This is the added value required originally to support Fishery knowledge experts in their work of aggregating data easily and fast.

The API provided to communicate with the FSDAS-server leaves space for future implementation of other kinds of clients, or web services wrapping the API methods.

The query capabilities have improved from the first prototype, especially with respect to query composition. While initially the generation of the query panel was static, now its rendering is subordinated to a lower level representation. This means that if in the future a new kind of query is required to be performed, and the graphical elements to input the values are missing, then an XML representation of the new query will drive the addition of the missing pieces when panel is created to be displayed.

As opposite to these positive sides, the FSDAS second prototype shows some shortcomings due to delays experienced during the development. The fact to align effort from many different kinds of competences around the same subject required a bootstrap period before every expert involved shared the same or aligned viewed on the matter. For instance the generation of the knowledge base could have included other data repositories, but the time dedicate to the analysis and understanding of the fishery XSD data schema and Fishery naming convention, as well as coding systems, took longer than expected. This unexpected issue stole room for investigation in other sources of data. The client GUI has kept the minimum set of widgets to satisfy the user requirements.

The effort planned in the remaining time frame of the project will be used to assess the response of Fishery expert when using this prototype of FSDAS. We expect to have positive feedback on all the issues tackled from the first version, and relevant comments on the accuracy of resource retrieval as result of data aggregation. In this process we will also gather feedback about the underlying ontologies, which might lead to adaptations and evolution of the ontologies. Since FSDAS is a server-based application, new versions of the ontology can be rolled out easily.

6 References

[AGMES] FAO. Agricultural Metadata Element Set. http://www.fao.org/aims/agmes_intro.jsp

[AIDA] http://www.fao.org/fi/figis/devcon/schema/3_6/aida.xsd

[ASFIS] L. Garibaldi, S. Busilacchi. ASFIS List of species for fishery statistic purposes. ftp://ftp.fao.org/docrep/fao/006/y7527t/y7527t00.pdf

[DagaEtAl08] Enrico Daga, Valentina Presutti and Alberto Salvati. http://ontologydesignpatterns.org and Evaluation WikiFlow.SWAP Workshop, volume 426 of CEUR Workshop Proceedings, CEUR-WS.org (2008)

[DC] Dublin Core Metadata Initiative. http://dublincore.org/

[EDC] Extended Dublin Core. http://dublincore.org/schemas/xmls/qdc/2003/04/02/dc.xsd

[Fsdic] FIGIS XML. List of elements. http://www.fao.org/fi/figis/devcon/diXionary/figisdoc3.5.html

[FISTAT] Fisheries and Aquaculture Department. Statistics. http://www.fao.org/fi/website/FIRektrieveAction.do?dom=topic&fid=16062

[FSschema] XML schema for Fisheries Fact Sheets. http://www.fao.org/fi/figis/devcon/schema/3_6/fi.xsd

[Gangemi&Presutti09] Gangemi A., Presutti V. “Ontology Design Patterns”, in Staab S. et al. (eds.): Handbook of Ontologies (2nd edition), Springer, 2009.

[KAON2] KAON2 – An infrastructure for managing OWL-DL, SWRL, and F-Logic ontologies. http://kaon2.semanticweb.org/

[ISO2] International Standard Organization (ISO). "Codes for the representation of names of countries and their subdivisions." ISO 3166-1 ALPHA-2: 1997 (E/F), International Organization for Standardization. Geneva, 1997 (2006).

[ISO3] International Standard Organization (ISO): ISO 3166 ALPHA-3, 2006.

[ISSCFG] International Standard Statistical Classification of Fishing Gear (ISSCFG) ftp://ftp.fao.org/FI/DOCUMENT/cwp/handbook/annex/AnnexM1fishinggear.pdf

[NeOn D2.2.3] Marta Sabou, Guadalupe Aguado de Cea, Mathieu d’Aquin, Enrico Daga, Holger Lewen, Elena Montiel-Ponsoda, Valentina Presutti, MariCarmenSuárez-Figueroa. D2.2.3: Methods and Tools for the Evaluation and Selection of Knowledge Components. Available from NeOn project web site: http://www.neon-project.org (2009).

[NeOn D2.3.1] Klaas Dellschaft et al. D2.3.1: Practical Methods to Support Collaborative ontology Design. Available from NeOn project web site: http://www.neon-project.org (2008).

[NeOn D2.5.1] V. Presutti, A. Gangemi, et al. D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies. NeOn Project Deliverable, available at http://www.neon-project.org (2008).

[NeOn D5.4.2] Mari-Carmen Suarez-Figueroa, Eva Blomqvist, et al. D5.4.2: Revision and extension of the NeOn methodology. Available from NeOn project web site: http://www.neon-project.org (2009).

[NeOn D7.1.1] User requirements specifications for the Fisheries ontology, knowledge tools and alert system. NeOn Deliverable, November 2006.

[NeOn D7.2.2] Revised/enhanced Fisheries ontologies. NeOn Deliverable, 2007. http://www.neon-project.org/web-content/images/Publications/neon_2007_d7.2.2.pdf

[NeOn D7.2.3] Caterina Caracciolo, Aldo Gangemi, Juan Heguiabehere, Valentina Presutti. D7.2.3: Initial Network of Fisheries Ontologies. Available from NeOn project web site: http://www.neon-project.org (2009).

[NeOn D7.5.1] Software architecture for the ontology-based Fisheries Stock Depletion Assessment System (FSDAS). NeOn Deliverable, August, 2007.

[NeOn D7.6.1] First prototype of the Fisheries Stock Depletion Assessment System (FSDAS). NeOn Deliverable, May, 2008.

[NeOn D7.7.1] Evaluation of the FSDAS first prototype and recommendations to research. NeOn Deliverable, September, 2008.

[M49] United Nations. UN Code (M49). http://unstats.un.org/unsd/methods/m49/m49alpha.htm.

[Presutti&Gangemi08] Presutti V., Gangemi A. “Content Ontology Design Patterns as practical building blocks for web ontologies”. In Spaccapietra S. et al. (eds.): Proceedings of ER2008, Barcelona, Spain, 2008.

7 Appendix 1: How to install FSDAS v2

7.1 FSDAS Client Installation

No installation of any software is required. Users just need to direct their web browsers to the following URL:

http://212.170.156.131:10000/FSDAS_web/fsdas.zul.

7.2 FSDAS Server Installation

1. Download server installation files from:

   http://secse.atosorigin.es:10000/FSDAS_web /FSDAS-ServerInstallationFiles.zip

2. The Server Installation file contains the different files, which deploy the FSDAS prototype v2. In the following steps we describe how to deploy it in your server:

   a. Install the knowledge base files

      - Unzip the `KB.zip`
      - Remember the KB path for the OntoBroker server configuration (section 7.3)

   b. Unzip the file `apache-tomcat-6.0.16.zip`

   c. Deploy the Web application (FSDAS_web.war) within the webapps folder of the Apache Tomcat.

   d. Start Apache Tomcat.

7.3 FSDAS Server Configuration

In order to properly configure the FSDAS server we need to set some OntoBroker configurations on the server side to specify where the location of the knowledge base or the license key.

1. Change the path of the knowledge base

   - open the OntoConfig.prp file (you can find it in configfiles.zip)
   - go to line 78 (H2.URL = jdbc:h2:file:kbase/edb)
   - replace "kbase" with the full path to your knowledge base folder, e.g. d:/stuff/kbase
   - I.e.: the line should in this case "H2.URL = jdbc:h2:file:d:/stuff/kbase/edb"

2. Change the location of configuration files

   - Put the configuration files (OntoConfig.prp and ob_prof.key.xml) in the same folder as your development application (Eclipse or Tomcat for example)

8 Appendix 2: How to use FSDAS v2?

This appendix describes basic steps for using the second prototype of the Fisheries Stock Depletion Assessment System. This quick guide is intended for end-users in order to make easier their interaction with the system and making it possible to evaluate this second prototype in terms of functionality and usability.

This guide goes into the most relevant use cases that allow the user to experience the interaction with the FSDAS web-based client.

8.1 Launching the application

The second version of the FSDAS has been released as a Web application. To start the application you should use a web browser, such as Internet Explorer or Mozilla Firefox. The application has been tested with the latest versions of these Web browsers.

Currently the application is hosted in a server located in ATOS premises. In this guide we therefore provide the current URL, which will be replaced with one in FAO servers once the deployment of the application is done. The current URL is

http://212.170.156.131:10000/FSDAS_web/fsdas.zul.

When the web page is loaded, the main desktop page is opened:

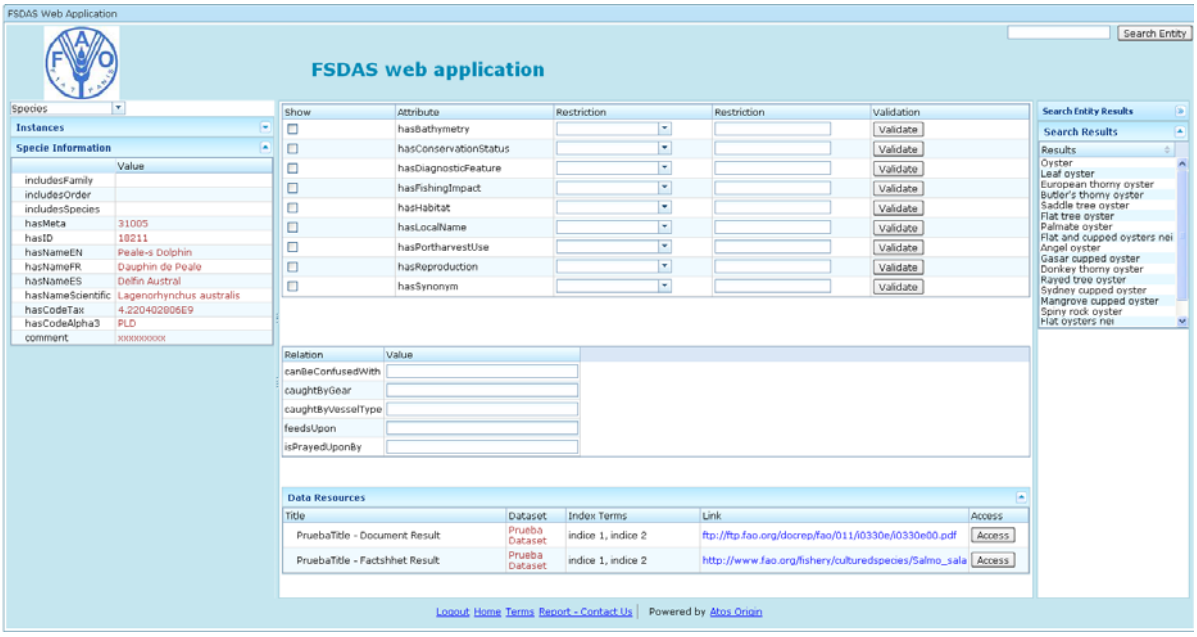


Figure 15: FSDAS web default perspective

The user interface is composed by the following widgets:

- Taxonomy Widget: It is located on the left hand side of the page. In this widget the taxonomy tree (i.e. species taxonomy, water area taxonomy...) is rendered. The data is extracted directly from FAO ontologies located on the server. The user is able select from a combo-box which taxonomy to visualize in the widget. When the user selects one node of the tree, the Entity Information Widget updates its content.

- Entity Information Widget: It is located on the left hand side of the page. This widget depicts the details of the item selected in the taxonomy widget. The details are extracted from the knowledge base, including object properties, attributes, annotations and its respective values.
- Search Widget: It is a textbox located at the top right of the page. The user may input free text in order to search ontological resources in the knowledge-base.
- Search Result Widget: It is the grid that shows the results after a search performed using the Search Widget. It is located in the right of the web page.
- Query Composition Widget: This is a panel that allows the user to dynamically compose queries to the knowledge base. The queries are formed dynamically (based on a configurable XML file located on the server). The possible queries are based on the ontology network stored in the data layer. When the user completes the required inputs and submits the query, the server side executes a query against the knowledge base using the facilities provided by the Data layer API.
- Query Results Widget: It is situated at the bottom centre of the page. This component uses a grid to display the query results returned from the query performed in the Query Composition Widget. Currently, the parameters shown in the grid are: Title of the resource, Dataset, index terms and the link to the resource. When the link is clicked, the resource is open in the assigned application (i.e. DOC files in Word, PDF in Acrobat Reader, etc.).

8.2 Execute Data Instance Query

ACTORS

- Fisheries scientist

DESCRIPTION

- User executes a query against the knowledge base (FAO ontology network). The query is built based on the ontologies (mainly application ontology) and the query choices are a fraction of the concepts and properties.

PRECONDITIONS

- All query values are selected
    - There is a data source selected within the taxonomy widget (Species, Water Area, Land Area...).
    - All fields belonging to the specific query are filled within Query Composition Widget.

TRIGGERING EVENT(s)

- Select the data source in the taxonomy widget.

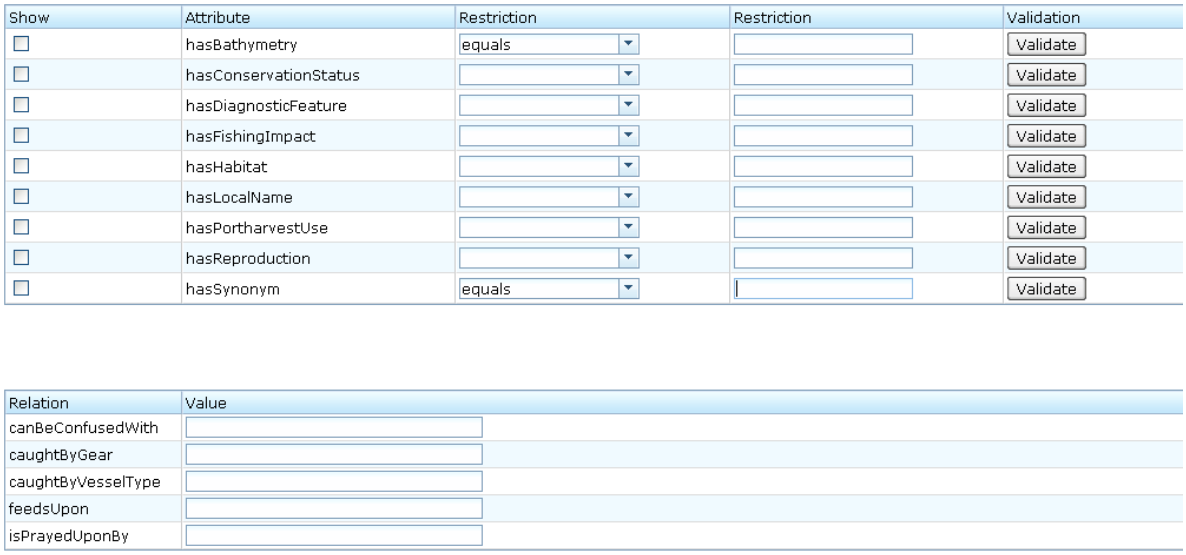


Figure 16: FSDAS web Query Composition widget

- User clicks the "validate" button in the same line than the attributes filled in. The query sent to the server side collects the attributes selected by the Show column and with filled restriction.

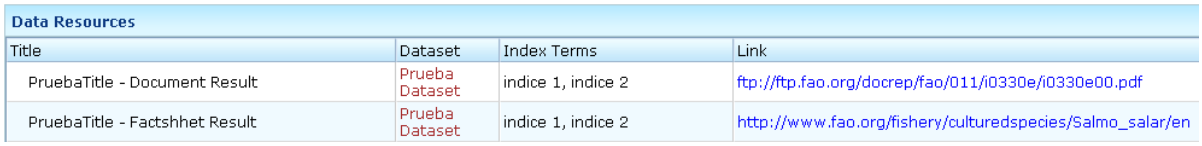
POST CONDITIONS

- System displays the query results.

FLOW OF EVENTS

a. BASIC FLOW

- User selects the data source in the combo of the taxonomy widget (i.e. species).
- User goes to the Query Composition widget and selects the attributes he would query.
    - For instance, user selects to ask by "*hasLocalName*" attribute.
    - User set the input keyword for the local name (i.e. "*shark*", "whale", "foca", "pez", etc.).
- User click on the Validate Button.
- System displays results in the Query Results Widget.
- User can sort results in the Query Results Widget by clicking any of the columns title.
- User can click on the url showed in each result and access/open the result resource provided by the system in the associated application.



Data Resources

Title	Dataset	Index Terms	Link
PruebaTitle - Document Result	Prueba Dataset	indice 1, indice 2	ftp://ftp.fao.org/docrep/fao/011/i0330e/i0330e00.pdf
PruebaTitle - Factshhet Result	Prueba Dataset	indice 1, indice 2	http://www.fao.org/fishery/culturedspecies/Salmo_salar/en

Figure 17: Query Results Widget

b. ALTERNATIVE FLOW: no results found

- System returns a no results alert.

RELATED USE CASES

- Any other Data Instance Query.

NOTES / ISSUES

For composed queries (various restrictions selected) it is compulsory to fill all fields of involved queries. The result will be the intersection of both results.

8.3 Execute Concept Search

ACTORS

- Fisheries scientist.

DESCRIPTION

- User executes Concept Search (Search for ontology content as concepts or data value).

PRECONDITIONS

- The ontologies are available at the server layer. These ontologies are loaded in Ontobroker and accessible via the API.

TRIGGERING EVENT(s)

Figure 18: FSDAS Concept Query perspective

- User clicks the “Search Entity” button in the same line than the textbox is filled in.

POST CONDITIONS

- System displays the results of the search.

FLOW OF EVENTS

a. BASIC FLOW

- User goes to the Search Widget at the top-right of the page.
- User sets the input keyword for the search (i.e. oyster, 27.0, North-Atlantic etc.).

- User clicks on the “Search Entity” Button
- System displays results in the Search Results Widget.

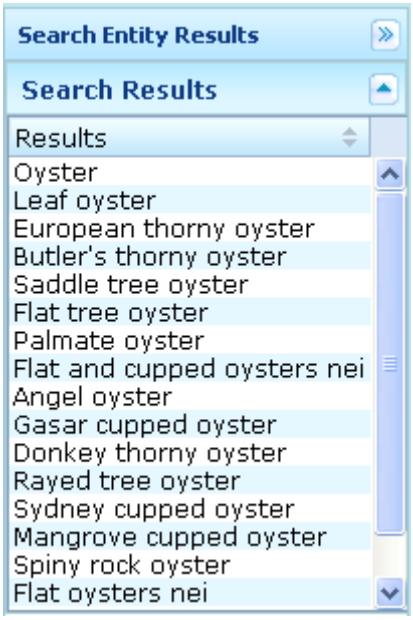


Figure 19: Concept Query 8 execution

b. ALTERNATIVE FLOW: no results found

- System returns a no results alert.

RELATED USE CASES

- Any other Concept / Data value Search.

NOTES / ISSUES

As a matter of example, the table below contains sample values for different concept search for testing purposes. It is especially advisable for users not familiarized with FAO ontologies. The last column of the table contains possible values that can be used as input in the textbox:

Table 3: Example of test query values

Entity	Attributes	DataValue
Species Group	With identifier	8502.0
		8505.0
		8507.0
	With scientific name	MOLLUSCA
		CRUSTACEA
		MAMMALIA
Species Order	With identifier	8004.0
		8006.0

Entity	Attributes	DataValue
		8008.0
	With scientific name	LAMNIFORMES
		ORECTOLOBIFORMES
		CHARACIFORMES
Species Family	With identifier	7002.0
		7004.0
		7008.0
	With scientific name	NOTOCHEIRIDAE
		CYPRINODONTIDAE
		LEPISOSTEIDAE
Water Area	With Englishname	Pac antarctc
		Ind antarctc
		EC Atlantic
	With FAO Code	58.0
		88.0

8.4 Taxonomy Widget & Entity Information Widget

ACTORS

- Fisheries scientist

DESCRIPTION

- User visualizes some selected FAO taxonomies (Land Areas, Water Areas, Species), based on the ontologies provided by FAO. The widget includes paging functionality in the branches of the taxonomy tree.

PRECONDITIONS

- The ontologies are available in the server side.
- The ontologies is previously loaded in OntoBroker and extracted its associated instance taxonomy in the server business logic.

TRIGGERING EVENT(s)

- The user can select which taxonomy visualize using the combo box inserted in the Taxonomy widget.

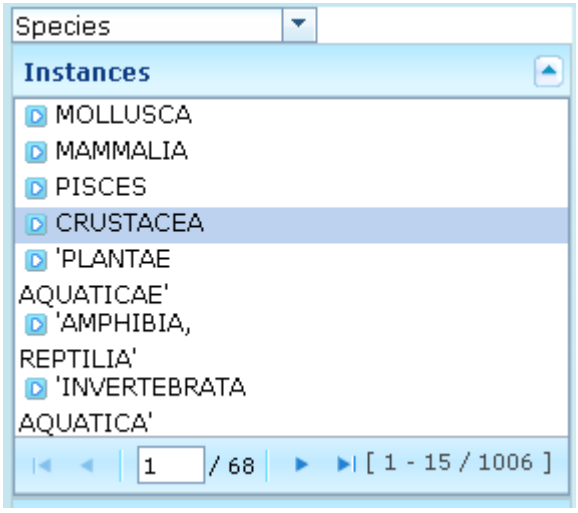


Figure 20: FSDAS web Instance Taxonomies Visualization

POST CONDITIONS

- System displays the instances taxonomy in the Taxonomy widget

FLOW OF EVENTS

a. BASIC FLOW

- User selects the ontology in the combo box.
- The ontology is visualized within the Taxonomy Widget.
- User clicks over the tree elements showed in the widget to expand or contract the different elements of the tree.
- The widget provides paging functionality when the taxonomy tree showed is bigger than 30 nodes.
- When the user double-clicks over any node of the taxonomy, the entity information widget update its content with the semantic information of the node extracted from the knowledge base.

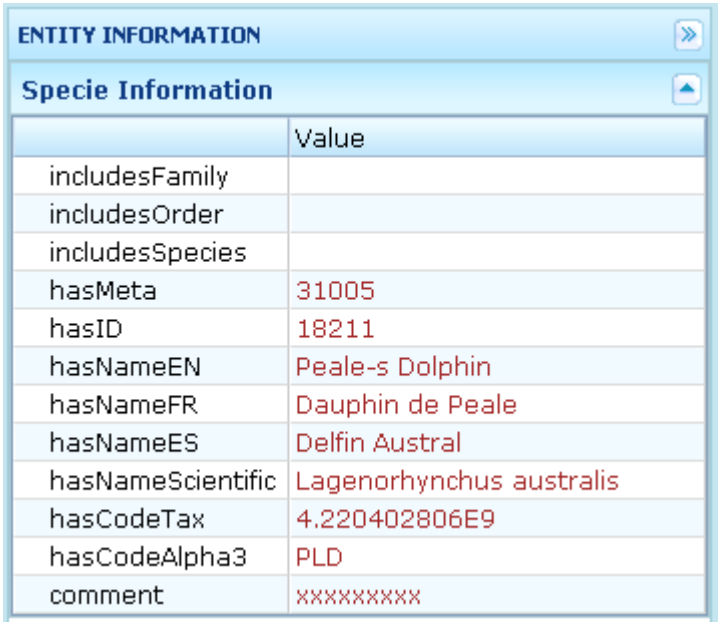


Figure 21: Entity Information Widget

b. ALTERNATIVE FLOW: no results found

- The ontology is not available
- The entity information of the node is not found.

RELATED USE CASES

- Any other Ontology visualization.
- Any other Entity Information visualization as View ontological resource annotation.